Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

Claims 1-7. (Canceled)

Claim 8. (Currently amended) A superscalar microprocessor for processing instructions, the microprocessor comprising:

an instruction fetch unit configured to fetch instructions from an instruction store according to a sequential program order;

a branch prediction circuit configured to provide a branch bias signal indicating whether a conditional branch controlled by a conditional branch instruction is predicted to be taken or not taken;

an instruction buffer coupled to receive fetched instructions from the instruction fetch unit and configured to buffer a plurality of fetched instructions, including an instruction selected according to the branch bias signal;

a plurality of functional units configured to execute instructions, thereby generating result data;

a register file including a plurality of entries configured to store data including result data generated by the plurality of functional units, wherein each of the plurality of entries is accessible by reference to a respective location in the register file;

a resource identifying circuit configured to concurrently identify execution resources for a first one and a second one of a plurality of buffered instructions, wherein the second one of the buffered instructions has a data dependency on the first one of the

buffered instructions, thereby making a plurality of instructions concurrently available for issue, wherein the identified execution resources for each of the available instructions includes a functional unit capable of executing the instruction;

a register rename circuit configured to provide references to locations in the register file for logical register references included with the plurality of buffered instructions;

an issue control circuit coupled to the resource identifying circuit and configured to concurrently issue more than one of the available instructions to the functional units for execution, based on availability of the identified execution resources for each instruction and availability of respective operands for each instruction in the referenced locations in the register file, without regard to the sequential program order;

a plurality of data routing paths coupled between the plurality of functional units and the register file and configured to concurrently transfer result data from more than one of the plurality of functional units to the register file; and

bypass control logic coupled to the plurality of data routing paths and configured to distribute result data from a first one of the plurality of functional units as operand data for another one or more of the plurality of functional units via an alternate data path that bypasses the register file, wherein distributing result data via the alternate data path occurs concurrently with transferring result data to the register file.

Claim 9. (Previously presented) The microprocessor of claim 8, wherein:

the plurality of functional units includes an integer functional unit and a floating-point functional unit; and

the bypass control logic is further configured such that an integer result from the integer functional unit is distributed to the floating-point functional unit via the alternate data path.

Claim 10. (Previously presented) The microprocessor of claim 8, wherein:

the plurality of functional units includes an integer functional unit and a floating-point functional unit; and

the bypass control logic is further configured such that a floating-point result from the floating-point functional unit is distributed to the integer functional unit via the alternate data path.

Claim 11. (Previously presented) The microprocessor of claim 8, further comprising:

operand data routing paths coupled between the register file and the functional units and configured to concurrently transfer operand data to more than one of the functional units.

Claim 12. (Previously presented) The microprocessor of claim 11, wherein the operand data routing paths transfer operand data directly from the register file to the functional units.

Claim 13. (Canceled)

Claim 14. (Currently amended) A method for processing instructions in a superscalar microprocessor, the method comprising:

fetching instructions from an instruction store according to a sequential program order;

predicting whether a conditional branch controlled by a conditional branch instruction included in the fetched instructions is taken or not taken;

buffering a plurality of fetched instructions, including an instruction selected according to the prediction, in an instruction buffer;

concurrently identifying execution resources for more than one of a plurality of buffered instructions, the identified execution resources for each of the more than one of the plurality of buffered instructions including a functional unit capable of executing the instruction;

providing references to locations in a register file for logical register references included with the plurality of buffered instructions, wherein the register file includes a plurality of entries, each of the plurality of entries being accessible by reference to a respective location in the register file;

concurrently making available for execution a plurality of instructions for which execution resources are identified and register file location references are provided;

concurrently issuing more than one of the plurality of available instructions for execution by a plurality of functional units, based on availability of the identified execution resources for each available instruction and availability of respective operands for each instruction in the referenced locations in the register file, without regard to the sequential program order;

executing the issued instructions in the plurality of functional units, thereby generating result data;

transferring the result data from the functional units to the register file; [[and]]

concurrently with said act of transferring, distributing the result data from a first one of the plurality of functional units as operand data for another one or more of the plurality of functional units via a bypass data path that bypasses the register file; and

retiring instructions according to the sequential program order.

15. (Previously presented) The method of claim 14, wherein:

the plurality of functional units includes an integer functional unit and a floating point functional unit; and

the act of distributing the result data includes distributing result data from the integer functional unit to the floating point functional unit via the bypass data path.

16. (Previously presented) The method of claim 14, wherein:

the plurality of functional units includes an integer functional unit and a floating point functional unit; and

the act of distributing the result data includes distributing result data from the floating point functional unit to the integer functional unit via the bypass data path.

17. (Previously presented) The method of claim 14, further comprising:

concurrently transferring operand data from the register file to more than one of the functional units via a plurality of operand data routing paths.

18. (Previously presented) The method of claim 17, wherein the operand data routing paths transfer operand data directly from the register file to the functional units.

19. (Canceled)

Claim 20. (Currently amended) The ~~microprocessor of claim 8, further comprising~~ A superscalar microprocessor for processing instructions, the microprocessor comprising:

an instruction fetch unit configured to fetch instructions from an instruction store according to a sequential program order;

a branch prediction circuit configured to provide a branch bias signal indicating whether a conditional branch controlled by a conditional branch instruction is predicted to be taken or not taken;

an instruction buffer coupled to receive fetched instructions from the instruction fetch unit and configured to buffer a plurality of fetched instructions, including an instruction selected according to the branch bias signal;

a plurality of functional units configured to execute instructions, thereby generating result data;

a register file including a plurality of entries configured to store data including result data generated by the plurality of functional units, wherein each of the plurality of entries is accessible by reference to a respective location in the register file;

a resource identifying circuit configured to concurrently identify execution resources for a plurality of buffered instructions, thereby making a plurality of instructions concurrently available for issue, wherein the identified execution resources for each of the available instructions includes a functional unit capable of executing the instruction;

a register rename circuit configured to provide references to locations in the register file for logical register references included with the plurality of buffered instructions;

an issue control circuit coupled to the resource identifying circuit and configured to concurrently issue more than one of the available instructions to the functional units for execution, based on availability of the identified execution resources for each instruction and availability of respective operands for each instruction in the referenced locations in the register file, without regard to the sequential program order;

a plurality of data routing paths coupled between the plurality of functional units and the register file and configured to concurrently transfer result data from more than one of the plurality of functional units to the register file;

bypass control logic coupled to the plurality of data routing paths and configured to distribute result data from a first one of the plurality of functional units as operand data for another one or more of the plurality of functional units via an alternate data path that bypasses the register file, wherein distributing result data via the alternate data path occurs concurrently with transferring result data to the register file; and

retirement control logic coupled to the register file and configured to concurrently retire a plurality of instructions according to the sequential program order.

Claim 21. (Previously presented) The microprocessor of claim 20, wherein the register file includes:

a temporary buffer having a first plurality of entries; and

a retired register array having a second plurality of entries;

and wherein the retirement control logic is further configured such that when an instruction is retired, corresponding result data is transferred from the temporary buffer to the retired register array.

Claim 22. (Canceled)

Claim 23. (Currently amended) The method of claim [[22]] 14, wherein the register file includes:

a temporary buffer having a first plurality of entries; and

a retired register array having a second plurality of entries;

and wherein the act of retiring an instruction includes transferring corresponding result data from the temporary buffer to the retired register array.

Claim 24. (Canceled)

Claim 25. (Currently amended) The microprocessor of claim 8, wherein the register rename circuit is further configured to concurrently provide references to locations in the register file for [[a]] the first one and [[a]] the second one of the plurality of the buffered instructions~~, wherein the second one of the buffered instructions has a data dependency on the first one of the buffered instructions~~.

Claim 26. (Previously presented) The microprocessor of claim 8, wherein the logical register references include at least one of a source register reference and a destination register reference.

Claim 27. (Previously presented) The method of claim 14, wherein concurrently identifying execution resources includes concurrently identifying execution resources for

a first one and a second one of the plurality of buffered instructions, wherein the second one of the instructions has a data dependency on the first one of the instructions.

Claim 28. (Previously presented) The method of claim 14, wherein providing references to locations in the register file includes concurrently providing references to locations in the register file for at least a first one and a second one of the buffered instructions, wherein the second one of the buffered instructions has a data dependency on the first one of the buffered instructions.

Claim 29. (Previously presented) The method of claim 14, wherein the logical register references include at least one of a source register reference and a destination register reference.

Claims 30-37 (Canceled).

Claim 38. (Currently amended) A data processing apparatus comprising a super scalar type microprocessor having a plurality of functional units that can execute instructions simultaneously, the microprocessor comprising:

a pre-fetch unit that pre-fetches a plurality of instructions from a memory in preparation for execution by one or more functional units, the plurality of instructions having a predetermined program order;

a branch prediction circuit configured to provide a branch bias signal indicating whether a conditional branch controlled by a conditional branch instruction is predicted to be taken or not taken;

a buffer that holds a plurality of instruction groups, including one or more instruction groups pre-fetched by the pre-fetch unit according to the branch bias signal;

a decoder that simultaneously decodes a plurality of instructions from an instruction group held in the buffer;

a register file including a plurality of registers used in the one or more functional units executing the plurality of decoded instructions, the plurality of registers including a temporary buffer that stores results from execution of instructions outside the predetermined program order;

a dependency check unit that checks for a dependency relation between the plurality of instructions output from the decoder, on the basis of use conditions stored in a register;

an instruction unit that allocates an instruction to a functional unit so that the instruction executes outside the predetermined program order after the instruction is judged by the dependency check unit not to be subject to restriction due to a dependency, wherein when executing instructions outside the predetermined program order, the microprocessor uses the temporary buffer; and

a retirement unit that specifies a register in which to store a result of executing the instruction outside the predetermined program order, wherein the retirement unit retires the instruction in program order after the instruction is completed, wherein when completing the instructions executed outside the predetermined program order, contents of the temporary buffer are written in a corresponding register to retire the instructions.

Claim 39. (Canceled).

Claim 40. (Previously presented) The data processing apparatus of claim 38, further comprising a memory from which the microprocessor accesses data through a system memory bus to allow execution of the instruction group.

Claims 41-42. (Canceled).

Claim 43. (Previously presented) A superscalar microprocessor for executing instructions, the microprocessor comprising:

an instruction fetch unit configured to fetch instructions from an instruction store according to a sequential program order; and

an instruction execution unit configured to concurrently receive a set of from 1 to a maximum number (N) of instructions from the instruction fetch unit, the instruction execution unit including:

an instruction buffer configured to store instruction information for each instruction received from the instruction fetch unit, wherein the instruction buffer has sufficient capacity to store the instruction information for at least twice the number N of instructions;

a register file comprising a plurality of temporary buffers and a plurality of retired registers, wherein the temporary buffers are arranged in a plurality of groups of temporary buffers, each group of temporary buffers including N of the temporary buffers;

renaming logic configured to concurrently establish an association between each instruction in a set of instructions concurrently received from the instruction fetch unit and a respective one of the temporary buffers in a selected

one of the groups of temporary buffers, wherein a position of each instruction within the set of instructions determines which one of the temporary buffers in the selected group of temporary buffers is associated with that instruction;

a plurality of functional units configured to execute instructions, thereby generating result data;

an issue control circuit configured to concurrently issue more than one of the instructions for which instruction information is stored in the instruction buffer to the functional units for execution, the issue control circuit being further configured to issue at least some of the instructions out of the sequential program order;

a plurality of data routing paths coupled between the functional units and the register file and configured to transfer result data from more than one of the functional units to the temporary buffers concurrently; and

retirement control logic coupled to the register file and configured to retire instructions according to the sequential program order, wherein the retirement control logic is further configured to concurrently retire all of the instructions in a set of instructions after all of the instructions in that set of instructions have completed.

Claim 44. (Previously presented) The superscalar microprocessor of claim 43 wherein the retirement control logic is further configured to concurrently retire all of the instructions in a set of instructions after all of the instructions in the set of instructions have completed and all instructions that precede the set of instructions according to the sequential program order have been retired.

Claim 45. (Previously presented) The superscalar microprocessor of claim 43 wherein the retirement control logic is further configured such that retiring all of the instructions in a set of instructions includes transferring the result data from the temporary buffers in the group of temporary buffers associated with the set of instructions to selected ones of the retired registers.

Claim 46. (Previously presented) The superscalar microprocessor of claim 45 wherein the temporary buffers are configured as a FIFO and wherein each group of temporary buffers corresponds to an entry in the FIFO.

Claim 47 (Previously presented) The superscalar microprocessor of claim 46 wherein the retirement control logic is further configured to advance the FIFO by one group of temporary buffers upon retirement of each set of instructions.

Claim 48. (Previously presented) The superscalar microprocessor of claim 43 wherein:

the instruction information stored in the instruction buffer includes an identifier of one of the retired registers; and

the retirement control logic is further configured such that retiring all of the instructions in the set of instructions includes transferring the result data for at least one of the instructions in the set of instructions from the temporary buffer associated with that instruction to the one of the retired registers identified by the retired register identifier stored for that instruction.

Claim 49. (Previously presented) The superscalar microprocessor of claim 43 wherein the instruction buffer is configured as a FIFO and wherein each entry in the FIFO has sufficient capacity to store instruction information for N instructions.

Claim 50. (Previously presented) The superscalar microprocessor of claim 43 wherein the instruction information stored in the instruction buffer includes information indicating an operation to be performed.

Claim 51. (Previously presented) The superscalar microprocessor of claim 43 wherein the instruction buffer has the capacity to store the instruction information for exactly twice the number N of instructions.

Claim 52. (Previously presented) The superscalar microprocessor of claim 43 wherein the number of temporary buffers is equal to the number of instructions for which the instruction buffer has the capacity to store the instruction information.

Claim 53. (Previously presented) The superscalar microprocessor of claim 43 wherein N is four.

Claim 54. (Previously presented) The superscalar microprocessor of claim 43 wherein the instruction execution unit further includes:

a decode circuit configured to concurrently decode all of the instructions in a set of instructions concurrently received by the instruction execution unit.

Claim 55. (Previously presented) The superscalar microprocessor of claim 54 wherein the decode circuit is disposed at a pipeline stage subsequent to a stage at which the instruction information for a set of instructions is stored into the instruction buffer.

Claim 56. (Previously presented) The superscalar microprocessor of claim 43 further comprising:

bypass control logic coupled to the plurality of data routing paths and configured to supply result data from one of the functional units as operand data for another one of the functional units via an alternate data path that bypasses the register file, wherein supplying the result data via the alternate data path occurs concurrently with transferring the result data to the temporary buffers.

Claim 57. (Previously presented) The superscalar microprocessor of claim 43 further comprising:

a register rename circuit configured to concurrently identify, for each instruction in a set of instructions concurrently received by the instruction execution unit, a register file entry corresponding to a source of an operand for the instruction.

Claim 58. (Previously presented) The superscalar microprocessor of claim 57 wherein the register rename circuit is further configured to concurrently identify a register file entry corresponding to a source of an operand for each of a first instruction and a second instruction in a set of instructions wherein the second instruction has a data dependency on the first instruction.

Claim 59. (Previously presented) The superscalar microprocessor of claim 43 wherein the instruction fetch unit is further configured to provide a new set of instructions to the instruction execution unit in response to a previous set of instructions being retired.

Claim 60. (Previously presented) The superscalar microprocessor of claim 43 wherein:

the instruction fetch unit includes a branch prediction circuit configured to detect a first conditional branch instruction among the instructions fetched from the instruction store and to predict whether a first conditional branch corresponding to the first conditional branch instruction will be taken or not taken, and

the instruction fetch unit is further configured to deliver instructions subsequent to the first conditional branch instruction to the instruction execution unit based on the prediction.

Claim 61. (Previously presented) The superscalar microprocessor of claim 60 wherein:

the branch prediction circuit is further configured to detect a second conditional branch instruction among the instructions fetched from the instruction store and to predict, prior to a determination of whether the first conditional branch was correctly predicted, whether a second conditional branch corresponding to the second conditional branch instruction will be taken or not taken, and

the instruction fetch unit is further configured to deliver instructions subsequent to the second conditional branch instruction to the instruction execution unit based on the prediction as to the second conditional branch.

Claim 62. (Previously presented) The superscalar microprocessor of claim 61 wherein the instruction fetch unit is further configured to continue delivering instructions to the instruction execution unit as long as the instruction buffer is not filled with sets of

instructions, regardless of the number of conditional branch instructions that have been previously delivered to the instruction execution unit but not retired.

Claim 63. (Previously presented) A method for executing instructions in a superscalar microprocessor, the method comprising:

fetching instructions from an instruction store according to a sequential program order;

concurrently delivering a set of from 1 to a maximum number (N) of fetched instructions to an instruction execution unit, wherein the instruction execution unit includes a register file comprising a plurality of temporary buffers and a plurality of retired registers, wherein the temporary buffers are arranged in a plurality of groups of temporary buffers, each group of temporary buffers including N of the temporary buffers;

storing instruction information for each instruction in the set of delivered instructions in an instruction buffer of the instruction execution unit, wherein the instruction buffer has sufficient capacity to store the instruction information for at least twice the number N of instructions;

concurrently establishing an association between each instruction in the set of instructions delivered by the instruction fetch unit and a respective one of the temporary buffers in a selected one of the groups of temporary buffers, wherein a position of each instruction within the set of instructions determines which one of the temporary buffers in the selected group of temporary buffers is associated with that instruction;

concurrently issuing more than one of the instructions for which instruction information is stored in the instruction buffer to a plurality of functional units, wherein at least some of the instructions are issued out of the sequential program order;

executing the issued instructions in the plurality of functional units, thereby generating result data;

concurrently transferring the result data from more than one of the plurality of functional units to the temporary buffers; and

concurrently retiring all of the instructions in the set of instructions after all of the instructions in the set of instructions have completed.

Claim 64. (Previously presented) The method of claim 63 wherein the act of concurrently retiring is performed after all of the instructions in the set of instructions have completed and all of the instructions that precede the set of instructions in the sequential program order have been retired.

Claim 65. (Previously presented) The method of claim 63 wherein the act of concurrently retiring all of the instructions in the set of instructions includes transferring the result data from the temporary buffers in the group of temporary buffers associated with the set of instructions to selected ones of the retired registers.

Claim 66. (Previously presented) The method of claim 65 wherein the temporary buffers are configured as a FIFO and wherein each group of temporary buffers corresponds to an entry in the FIFO, the method further comprising:

advancing the FIFO by one group of temporary buffers after retiring the set of instructions.

Claim 67. (Previously presented) The method of claim 63 wherein:

the instruction information stored in the instruction buffer includes an identifier of one of the retired registers; and

the act of concurrently retiring all of the instructions in the set of instructions includes transferring the result data for at least one of the instructions in the set of instruction from the temporary buffer associated with the at least one of the instructions to the one of the retired registers identified by the retired register identifier stored for the at least one of the instructions.

Claim 68. (Previously presented) The method of claim 63 wherein the instruction buffer is configured as a FIFO and wherein each entry in the FIFO has sufficient capacity to store instruction information for N instructions.

Claim 69. (Previously presented) The method of claim 63 wherein the instruction information stored in the instruction buffer includes information indicating an operation to be performed.

Claim 70. (Previously presented) The method of claim 63 wherein the instruction buffer has the capacity to store the instruction information for exactly twice the number N of instructions.

Claim 71. (Previously presented) The method of claim 63 wherein the number of temporary buffers is equal to the number of instructions for which the instruction buffer has the capacity to store the instruction information.

Claim 72. (Previously presented) The method of claim 63 wherein N is four.

Claim 73. (Previously presented) The method of claim 63 further comprising:

concurrently decoding all of the instructions in the set of instructions.

Claim 74. (Previously presented) The method of claim 73 wherein the act of concurrently decoding takes place subsequently to storing the instruction information for the set of instructions into the instruction buffer.

Claim 75. (Previously presented) The method of claim 63 further comprising:

supplying result data from one of the functional units as operand data for another one of the functional units via an alternate data path that bypasses the register file, wherein supplying the result data via the alternate data path occurs concurrently with transferring the result data to the temporary buffers.

Claim 76. (Previously presented) The method of claim 63 further comprising:

concurrently identifying, for each instruction in the set of instructions, a register file entry corresponding to a source of an operand for the instruction.

Claim 77. (Previously presented) The method of claim 76 wherein the act of concurrently identifying is performed regardless of whether one of the instruction in the set of instructions has a data dependency on another of the instructions in the set of instructions.

Claim 78. (Previously presented) The method of claim 63 further comprising:

delivering a new set of instructions to the instruction execution unit in response to the set of instructions being retired.

Claim 79. (Previously presented) The method of claim 63 further comprising:

detecting a first conditional branch instruction among the instructions fetched from the instruction store;

predicting whether a first conditional branch corresponding to the first conditional branch instruction will be taken or not taken; and

thereafter fetching one or more additional instructions from the instruction store based on the prediction.

Claim 80. (Previously presented) The method of claim 79 further comprising:

detecting a second conditional branch instruction among the additional instructions fetched from the instruction store based on the prediction as to the first conditional branch;

predicting, prior to a determination of whether the first conditional branch was correctly predicted, whether a second conditional branch corresponding to the second conditional branch instruction will be taken or not taken; and

fetching one or more further instructions from the instruction store based on the prediction as to the second conditional branch.

Claim 81. (Previously presented) The method of claim 80 wherein the act of concurrently delivering a set of fetched instructions to the instruction execution unit is repeated as long as the instruction buffer is not filled with sets of instructions, regardless of the number of conditional branch instructions that have been previously delivered to the instruction execution unit but not retired.